Il Business BASIC Apple

Biblioteca Linguaggi e programmi di utilita'

Il BASIC avanzato dell'Apple *III* per le applicazioni professionali

Il Business BASIC Apple per il Sistema di Elaborazione Apple III è una delle più potenti versioni di BASIC che siano mai state sviluppate per un microelaboratore. Progettato per quanti desiderano la flessibilità, la facilità d'impiego, e la potenza del popolare linguaggio BASIC, il Business BASIC Apple incorpora un certo numero di caratteristiche innovative che possono essere impiegate per soddisfare ad alcuni dei più esigenti requisiti nella programmazione per affari e scientifica.

Con il Business BASIC Apple, i programmatori possono indirizzare più di 70 Kbytes di memoria estesa — il più esteso spazio di lavoro disponibile su qualsiasi elaboratore personale. In aggiunta, la speciale rappresentazione dei dati a 64 bit, 18 digit, del Business BASIC Apple riesce a gestire i più difficili compiti di ragioneria con una precisione «al centesimo».

Per la produzione dei resoconti, il Business BASIC Apple offre alcune delle più complete e flessibili possibilità ritrovabili in qualsiasi versione di BASIC — persino in quelle impiegate negli elaboratori di grossi sistemi. Il BASIC per Affari Apple è anche caratterizzato da una gestione avanzata degli archivi, con formati flessibili per adattarsi alle richieste dell'applicazione; rappresentazione dei dati con interi binari a 16 bit, e a virgola mobile a 32 bit; nomi delle variabili a 64 caratteri (tutti i caratteri significativi) a scopo di documentazione; facile accesso ad un' ampia varietà di periferiche o dispositivi ausiliari collegabili; e un efficiente metodo di debugging.

Se preferite programmare in BASIC, ma avete bisogno di più potenza, flessibilità, e precisione di quanta la maggior parte dei BASIC vi possano offrire indirizzatevi al Business BASIC per l'Apple III — un linguaggio avanzato per un sistema di elaborazione avanzato. Il Business BASIC Apple viene fornito con tutte le configurazioni di sistema Apple III.

Vantaggi

Business BASIC Apple...

- *Consente una più facile generazione dei resoconti, in conseguenza delle sue capacità di uscita flessibili, estese e formattate, e delle versatili istruzioni PRINT USING e IMAGE...*
- *Aumenta l'efficienza del sistema, dato l'ampio spazio di lavoro (oltre 70 Kbytes) accessibile all'utente, che consente di tenere in memoria programmi molto lunghi...*
- *Aggiunge flessibilità alla programmazione, poichè vi consente di tenere aperti archivi multipli (fino a 10) simultaneamente, praticamente senza limitazione delle dimensioni...*
- *Vi consente di visualizzare e calcolare dati di contabilità finanziaria con precisione «al centesimo», mediante l'impiego di una rappresentazione dei dati an 64 bit, 18 cifre e di speciali funzioni...*
- *Accelera la registrazione su disco degli archivi, poichè le funzioni incorporate consentono ai programmi una lettura e una registrazione più efficiente degli archivi...*
- *Semplifica il controllo di input/output del sistema (I/0), poichè con poche parole chiave riesce possibile controllare un'ampia varietà di periferiche, nonchè le routines in linguaggio macchina e le funzioni grafiche...*
- *Aiuta nello sviluppo e nel debugging di programmi complessi, in quanto la sua istruzione opzionale ELSE e i listings inseriti automaticamente incoraggiano l'approccio strutturato alla programmazione.*

Il Business BASIC Apple - Un'occhiata ravvicinata

Il Business BASIC Apple è un linguaggio a carattere generale, per la soluzione di problemi, progettato specificamente per applicazioni commerciali e scientifiche, che richiede : 1) Un ambiente per il debugging di facile impiego; 2) Operazioni rapide e di alta precisione – come nella contabilità – su numeri in un intervallo di $\pm 10^{18}$; 3) Un elevato grado di formattazione dei dati per la stampa dei resoconti; 4) Una maggiore flessibilità per le operazioni 1/0 sugli archivi. Le potenti caratteristiche del Business BASIC Apple forniscono al programmatore molte possibilità avanzate e vantaggi.

Esteso spazio di memoria programma utente
In un Sistema di Elaborazione Apple*///* con 128 Kbytes di RAM, il Business BASIC Apple vi offre un ampio spazio di lavoro di 70 Kbytes – più di quanto sia disponibile su qualsiasi altro elaboratore personale BASIC. Ciò significa che potete scrivere grossi programmi con maggiore facilità e li potete far girare in modo più efficiente. Il tempo di accesso è ridotto, in quanto si possono tenere in memoria dei grossi archivi invece che su dischi. Inoltre si possono impiegare ampie zone per la grafica ad elevata risoluzione, senza temere di restringere in modo eccessivo la dimensione del programma.

Flessibile formattazione dei risultati
Le versatili istruzioni PRINT USING e IMAGE del Business BASIC Apple vi consentono di impiegare una varietà di stringhe formato per la preparazione dei resoconti. Le specifiche sono estremamente flessibili e includono categorie di notazione a stringhe, letterali, digitali, scientifica e di ingegneria.

Gestione avanzata degli archivi
Nel Business BASIC, la gestione degli archivi viene effettuata per mezzo di riferimenti numerati di archivio definiti nell'ambito del programma. E' possibile far effettuare operazioni a queste definizioni di archivio con il semplice impiego di termini generici. Per esempio, per stampare si richiama il termine generico «stampante» nell'ambito del programma. Il Business BASIC Apple lavora con il Sistema Operativo Sofisticato (SOS) dell'Apple *///* in modo da gestire in modo automatico i dettagli, includendo la stampante che viene indirizzata (se il vostro sistema ne ha più di una), il suo indirizzo 1/0 ecc. Per maggiore convenienza dell'utente, gli archivi su disco sono individuati mediante il nome di identificazione, sotto identificazione e archivio, senza riferimento al dispositivo di registrazione nel quale l'archivio risiede.

Nomi lunghi per le variabili
Il BASIC Apple vi consente di impiegare nomi per le variabili (fino a 64 caratteri di lunghezza) a scopo di documentazione, con caratteri tutti significativi. Inoltre, poichè gli spazi vengono impiegati come delimitatori, è consentito introdurre parole chiave BASIC nei nomi delle variabili, consentendo in tal modo una flessibilità ancora maggiore.

Caratteristiche BASIC «Strutturate»
Con il BASIC per Affari Apple, risulta possibile inserire l'istruzione ELSE alla normale istruzione IF... THEN del BASIC. E le vostre clausole ELSE possono a loro volta contenere altre costruzioni IF... THEN... ELSE. Il BASIC per Affari Apple include anche una caratteristica di LISTing che inserisce automaticamente i contenuti dei cicli FOR... NEXT. Queste caratteristiche BASIC «strutturate» incoraggiano l'utente ad impiegare un approccio più logico, strutturato, nello sviluppo dei loro programmi, e sono di notevole ausilio nello sviluppo e nel debugging dei programmi complessi.

Potente interfaccia per il linguaggio assembler
Poichè il Business BASIC Apple gira nell'ambiente SOS, voi non dovete preoccuparvi per la gestione della memoria, allocazione dei buffer, o spostamento degli archivi. Tale libertà vi farà risparmiare un sacco di tempo di programmazione.

Inoltre, il Business BASIC Apple è collegato con il SOS tramite un potente meccanismo di interfaccia Invoke/Perform. Se per

esempio avete creato delle routines in linguaggio macchina impiegando l'assemblatore su disco, potrete specificare la routine assemblata per nome in un programma Business BASIC. Una volta che le routines siano menzionate nell'istruzione «INVOKE», il Business BASIC Apple lavora con il SOS per trovare una residenza in memoria per le routines, e per fissare – come punti di entrata nel modulo interconnesso risultante – un qualunque nome di funzione o procedura menzionato nelle routines. Tutto quello che si deve fare è di eseguire (PERFORM) la routine precedentemente invocata – specificando le variabili che devono essere passate – e il Business BASIC Apple e il SOS gestiranno automaticamente tutti i dettagli operativi.

Il Business BASIC Apple rende la vita molto più facile ai programmatori espandendo le possibilità di sistema, riducendo il tempo di sviluppo del programma, ed aggiungendo una maggiore flessibilità alle funzioni di formattazione dei dati e di gestione archivi.

Specifiche Tecniche

Variabili

- 64 caratteri (max), tutti significativi
- Variabili riservate :
  ERR, KBD, EOF, VPOS, ERRLIN, HPOS, FRE, TIME$, DATE$, PREFIX$
- Rappresentazione dei dati
  Interi binari a 16 bit (da – 32768 a + 32767)
  Interi binari a 64 bit (± 9223372036854775807 o $2^{63} - 1$)
  Binari a 32 bit a virgola mobile (± 10^{38} con precisione a 6 digit)
  Stringhe di caratteri (da 0 a 255 caratteri, gestione dinamica).

Variabili multiple alfanumeriche e numeriche indicizzate a partire da 0, senza limiti dimensionali).

Operatori

- Generali : +, –, *, /, DIV, mod.

(Nota : DIV e MOD solo per operazioni su interi lunghi).

- Operatori logici binari :

AND, OR, =, <, >, <>, ><, >=, =<, =>.

- Operatore unario logico :

NOT

- Operatore alfanumerico

+ (concatenazione).

Istruzioni

(Nota : Nessuna istruzione o elenco di istruzioni può superare i 254 caratteri, includendo i delimitatori).

LET (opzionale)
REM
GOTO
IF ...GOTO
IF ...THEN
IF ...elenco istruzioni : ELSE elenco istruzioni
FOR variabile di controllo = espressione TO espressione
STEP espressione
NEXT variabile di controlo (,altra variabile di controllo)
GOSUB
RETURN
POP
ON espressione GOTO
ON espressione GOSUB
ON ERR
ON KBD
ON EOF#
OFF ERR
OFF KBD
OFF EOF#
RESUME

Istruzioni di utilità :

NEW	LOAD	STOP
CLEAR	SAVE	END
FRE	DELETE	CONT

PREFIX$	RUN	CHAIN nome programma, numero di linea

Funzioni definite dall'utente :
FN nome funzione (argomento)
DEF FN nome funzione (argomento) = espressione

Debugging :
TRACE
NOTRACE

Cursore e schermo :

LIST	VPOS	HOME	NORMAL
DEL	HPOS	INVERSE	TEXT

Funzioni alfanumeriche, numeriche e di archivio :

LEN	TEN	CONV	TAN	SQR
STR$	MID$	CONV%	ATN	EXP
VAL	LEFT$	CONV&	INT	LOG
CHR$	RIGHT$	CONV$	RND	TYP
ASC	SUB$	SIN	SGN	REC
HEX$	INSTR	COS	ABS	

Istruzioni per dati residenti di programma :
DATA
READ
RESTORE

Istruzioni e funzioni sui codici macchina :
INVOKE
PERFORM
EXFN
EXFN%

I/O di archivio :
CATALOG
DELETE
RENAME
LOCK
UNLOCK
CREATE
OPEN# numero di archivio (AS INPUT, AS OUTPUT, AS EXTENSION)
CLOSE# numero di archivio
CLOSE
INPUT# numero di archivio, numero di registrazione
OUTPUT# numero di archivio
PRINT# numero di archivio, numero di registrazione
PRINT# numero di archivio, numero di registrazione USING
READ# numero di archivio, numero di registrazione
WRITE# numero di archivio, numero di registrazione.

1/0 di console :
INPUT
GET
TAB
SPC
SCALE
PRINT
PRINT USING
IMAGE specificazione (i)

Specificazioni alfanumeriche :
A riserva una posizione per carattere giustificato a sinistra
C riserva una posizione per carattere giustificato al centro
R riserva una posizione per carattere giustificato a destra

Specifiche letterali :
X stampa uno spazio
/ stampa un ritorno carrello/avanzamento linea
«letterale» stampa tutto quanto tra virgolette

Specificazioni digitali :
riserva una cifra numerica, gli zeri che precedono vengono soppressi

Z riserva una cifra numerica, gli zeri che precedono vengono stampati
& riserva una cifra numerica o virgula
(la virgola copre tre cifre numeriche)
. riserva una posizione per il punto decimale
+ riserva una posizione per il segno
– riserva una posizione per il segno (se negativo)
$ riserva una posizione per il segno di dollaro
++ copertura asterisco
++ segno mobile
–– segno mobile (se negativo)
$$ segno di dollaro mobile

Specificazione per la notazione scientifica :
E riserva una posizione per l'esponente (potenza di dieci)

Specificazione per la notazione ingegneristica :
lo stesso che per la notazione scientifica, tranne che l'esponente è sempre un multiplo di tre.